Operators in Python

1.Arithmetic Operators

- Used to perform mathematical operations.

Operators	Meaning	Example	Result
+	Addition	4 + 2	6
–	Subtraction	4 – 2	2
*	Multiplication	4 * 2	8
/	Division	4 / 2	2
%	Modulus operator to get remainder in integer division	5 % 2	1
**	Exponent	5**2 = 5^2	25
//	Integer Division/ Floor Division	5//2 -5//2	2 -3

1. Addition (+) : The process of combining two or more numbers to get a total or a sum.

Example: 2 + 3 = 5

2. Subtraction (-) : The process of finding the difference between two numbers.

Example: 5 - 2 = 3

3. Multiplication (*) : The process of repeated addition of a number.

Example: 4 * 5 = 20 (4 added 5 times)

4. Division (/) : The process of sharing a quantity into equal parts or groups.

Example: 10 / 2 = 5 (10 divided into 2 equal parts)

5. Modulus (%) : The process of finding the remainder when one number is divided by another.

Example: 17 % 5 = 2 (remainder when 17 is divided by 5)

```
a = 10
b = 3

print(a + b)    # Addition: 13
print(a - b)    # Subtraction: 7
print(a * b)    # Multiplication: 30
print(a / b)    # Division: 3.333...
print(a // b)   # Floor Division: 3
print(a % b)    # Modulus: 1
print(a ** b)   # Exponentiation: 1000
```

2.Assignment Operators

- Used to assign values to variables.

Operator	Operation	Equivalent to
=	num = 5	num = 5
+=	num+=5	num = num+5
-=	num-=5	num = num-5
*=	num*=5	num = num*5
/=	num/=5	num = num/5
%=	num%=5	num = num%5

BeginnersBook.com

1. Assignment (=): An operator that assigns a value to a variable.

Example: x = 5 (assigns 5 to x)

2. Addition assignment (+=): An operator that adds a value to a variable and assigns the result.

Example: x += 3 (equivalent to x = x + 3)

3. Subtraction assignment (-=): Subtracts a value from a variable and assigns the result.

Example: x -= 2 (equivalent to x = x - 2)

4. Multiplication assignment (*=): Multiplies a variable by a value and assigns the result.

Example: x *= 4 (equivalent to x = x * 4)

5. Division assignment (/=): Divides a variable by a value and assigns the result.

Example: x /= 2 (equivalent to x = x / 2)

6. Modulus assignment (%=): Computes the remainder of dividing a variable by a value and assigns the result.

Example: x %= 5 (equivalent to x = x % 5)

```
x = 5
x += 2     # x = x + 2 → 7
print(x)

x -= 1     # x = x - 1 → 6
print(x)

x *= 3     # x = x * 3 → 18
print(x)
```

```
x /= 2     # x = x / 2 → 9.0
print(x)

x //= 2    # x = x // 2 → 4.0
print(x)

x %= 3     # x = x % 3 → 1.0
print(x)
```

3.Comparison Operators

- Used to compare values.

Operator	Meaning
== (double equal to)	Equal to
<	Less than
>	Greater than
!=	Not equal to
<=	Less than or equal to
>=	Greater than or equal to

1. Equal to (==): A comparison operator that checks if two values are equal.

Example: 2 == 2 (true)

2. Not equal to (!=): A comparison operator that checks if two values are not equal.

Example: 2 != 3 (true)

3. Greater than (>): A comparison operator that checks if one value is greater than another.

Example: 5 > 3 (true)

4. Less than (<): A comparison operator that checks if one value is less than another.

Example: 2 < 4 (true)

```
a = 5

b = 3

print(a == b)   # False
```

```
print(a != b)   # True

print(a > b)    # True

print(a < b)    # False

print(a >= b)   # True

print(a <= b)   # False
```

4.Logical Operators

- Used to combine conditional statements.

Operator	Meaning	Example	Result
and	Logical and	(5<2) and (5>3)	False
or	Logical or	(5<2) or (5>3)	True
not	Logical not	not (5<2)	True

1. And (&&) : A logical operator that combines two conditions and returns true if both are true.
   Example: true && true (true)
2. Or (||) : A logical operator that combines two conditions and returns true if at least one is true.
   Example : true || false (true)
3. Not (!) : A logical operator that reverses the truth value of a condition.
   Example: !true (false)

```
x = True

y = False

print(x and y)  # False

print(x or y)   # True

print(not x)    # False
```